I.K.G Punjab Technical University, Kapurthala

# Department of Computer Science Engineering

**LAB MANUAL**

**Internet of Things Lab BTCS 609-18**



# Bachelor of Computer Engineering (Computer Science Engineering)

**Submitted To: Submitted By:**

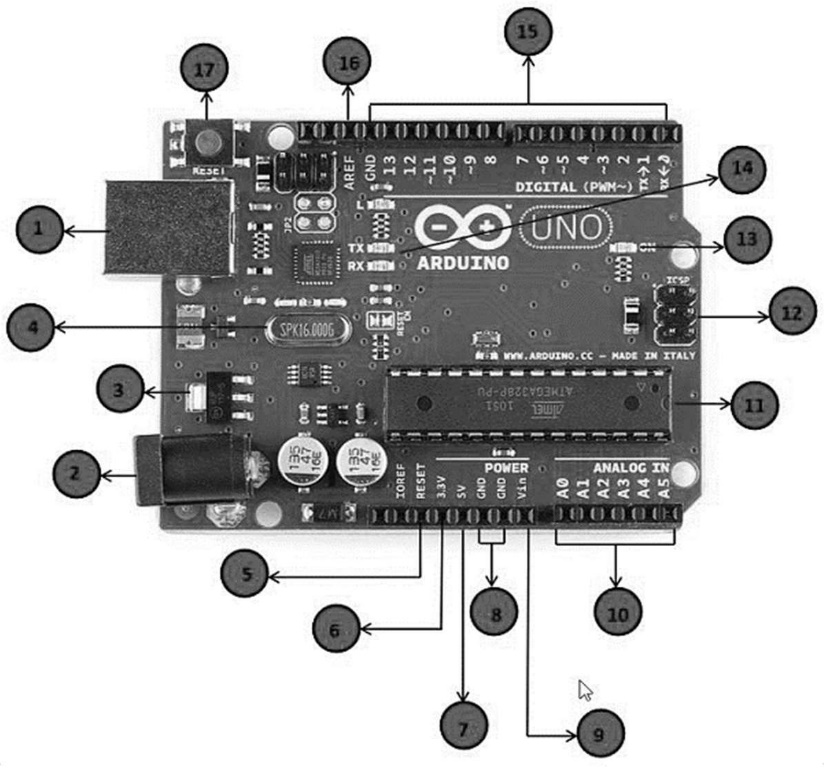
**Ms. Jasmine Attri Anupam Sharma**

**Roll no-2022939**

**COE A , 6th Sem**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Tasks** | **Page No.** | **Remarks** |
| **01.** | Familiarization with Arduino/Raspberry Pi and perform necessary software installation. |  |  |
| **02.** | To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds. |  |  |
| **03.** | To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push  button is pressed or at sensor detection. |  |  |
| **04.** | To interface DHT22 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings. |  |  |
| **05.** |  |  |  |
| **06.** | To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it. |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |



## TASK – 1

### Aim:- Familiarization with Arduino/Raspberry Pi and perform necessary software installation.

#### Introduction to Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

#### Pin Description of Arduino Uno Board



**Power USB**

Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).

**Power (Barrel Jack)**

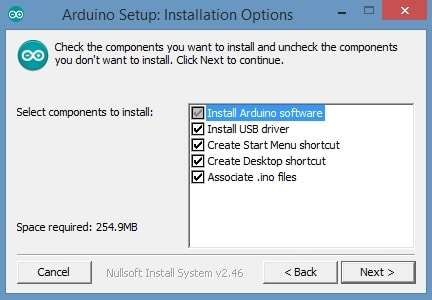
Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
|  |  | **Voltage Regulator**  The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements. |  |
|  | **Crystal Oscillator**  The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz. |  |
|  | **Arduino Reset**  You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5). |  |
|  | **Pins (3.3, 5, GND, Vin)**  3.3V (6) − Supply 3.3 output volt 5V (7) − Supply 5 output volt  Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.  GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.  Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply. |  |
|  | **Analog pins**  The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor. |  |
|  | **Main microcontroller**  Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet. |  |
|  | **ICSP pin**  Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus. |  |
| P a g e 4 | 15 | | |  |

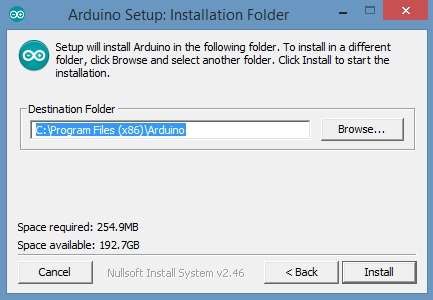
|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
|  |  | **Power LED indicator**  This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection. |  |
|  | **TX and RX LEDs**  On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process. |  |
|  | **Digital I/O**  The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled “~” can be used to generate PWM. |  |
|  | **AREF**  AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins. |  |
| * **Arduino IDE / Software installation steps: -**  1. Visit <http://www.arduino.cc/en/main/software>to download the latest Arduino IDE version for your computer’s operating system. 2. Save the .exe file to your hard drive. 3. Open the .exe file. 4. Click the button to agree to the licensing agreement:   P a g e 5 | 15 | | |  |



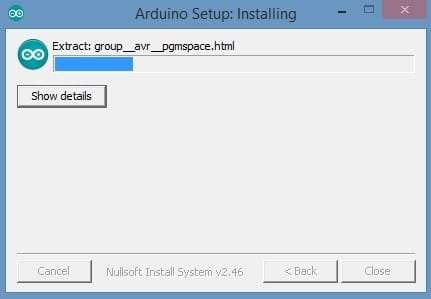
1. Decide which components to install, then click “Next”:



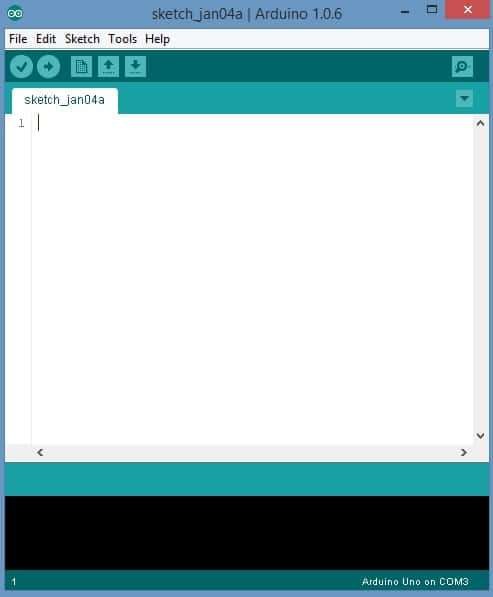
1. Select which folder to install the program to, then click “Install”:



1. Wait for the program to finish installing, then click “Close”:



1. Now find the Arduino shortcut on your Desktop and click on it. The IDE will open up and you’ll see the code editor:



## Task :-2

### Aim:-To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.

* **Apparatus :-** Beadboard , Jumper Wires , LED , Arduino UNO board , Data cable and Arduino IDE software.
* **Introduction :-** To turn ON an LED , the Arduino pin will send the high signal to the push button and when we will push the button then the sigh signal will delivered to the LED terminal that is (Cathode) and finally LED will glow.

#### CODE

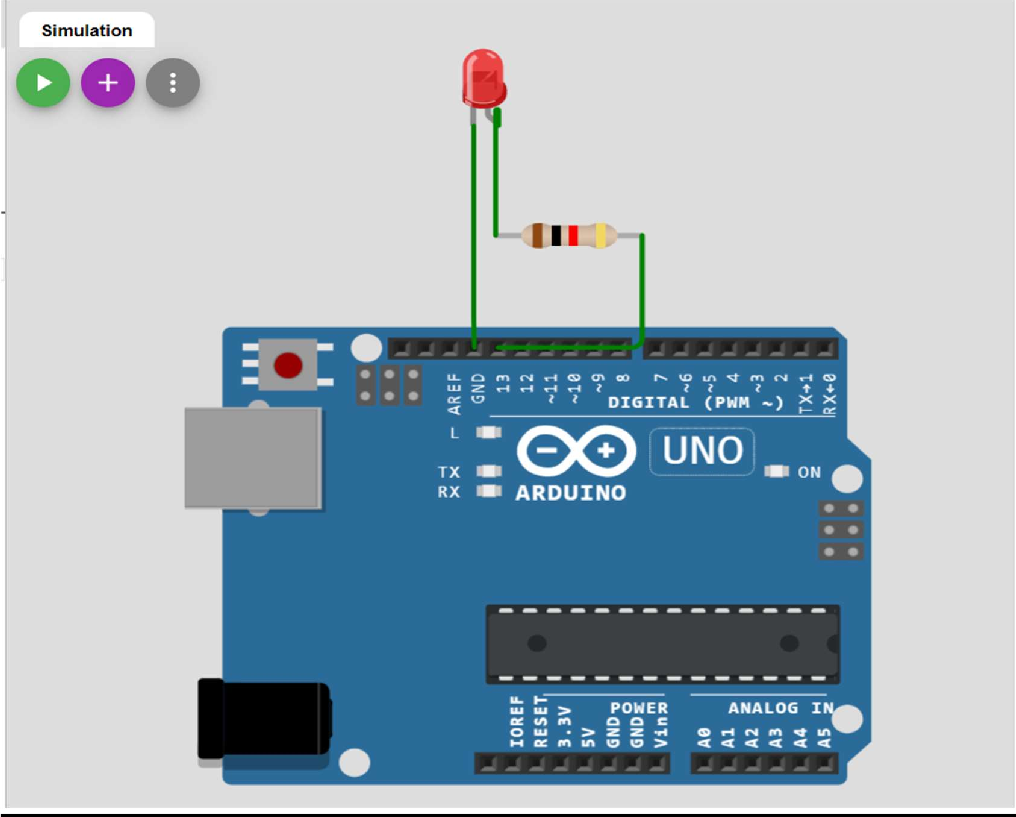
**void setup() { pinMode(LED\_BUILTIN, OUTPUT);**

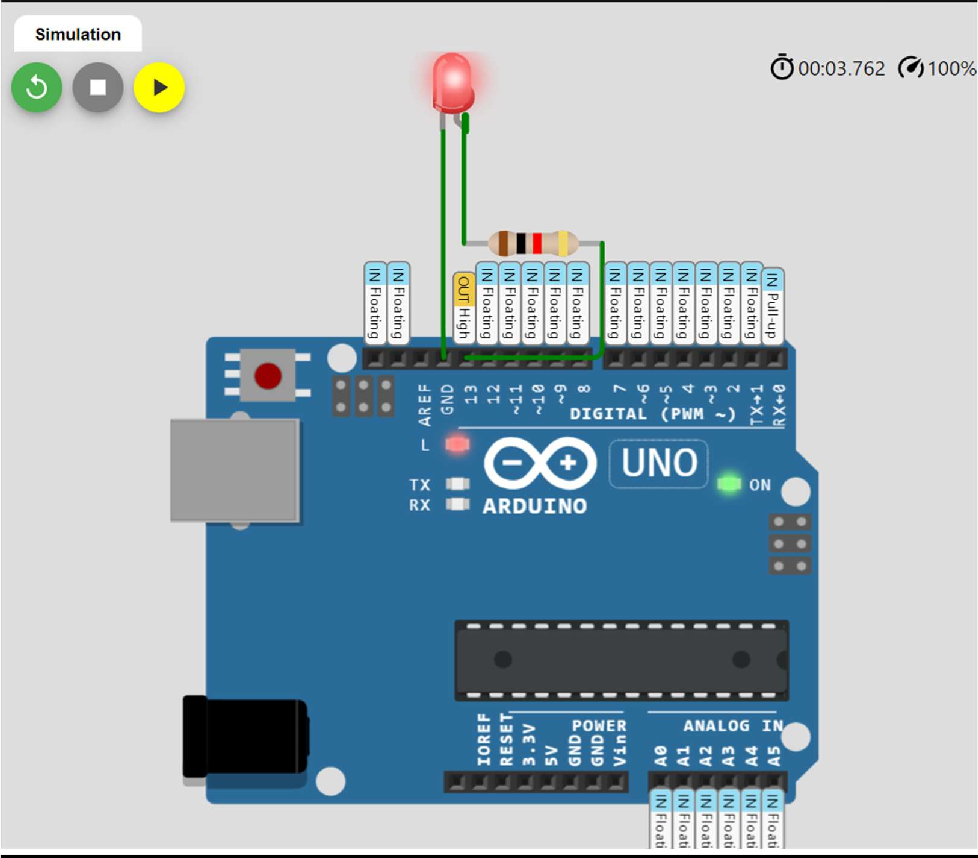
**}**

**void loop() { digitalWrite(LED\_BUILTIN, HIGH); delay(1000); digitalWrite(LED\_BUILTIN, LOW); delay(2000);**

**}**

**OUTPUT:-**





## Task :- 03

### Aim :- To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.

* **Apparatus :-** Beadboard , Jumper Wires , Push Button , Arduino UNO board , Data cable and Arduino IDE software.
* **Introduction:-** Push buttons connect two points in a circuit when you press them. This example turns on one led when the button pressed once, and off when release.

#### CODE :-

void setup() { pinMode(6, INPUT);

pinMode(13, OUTPUT);

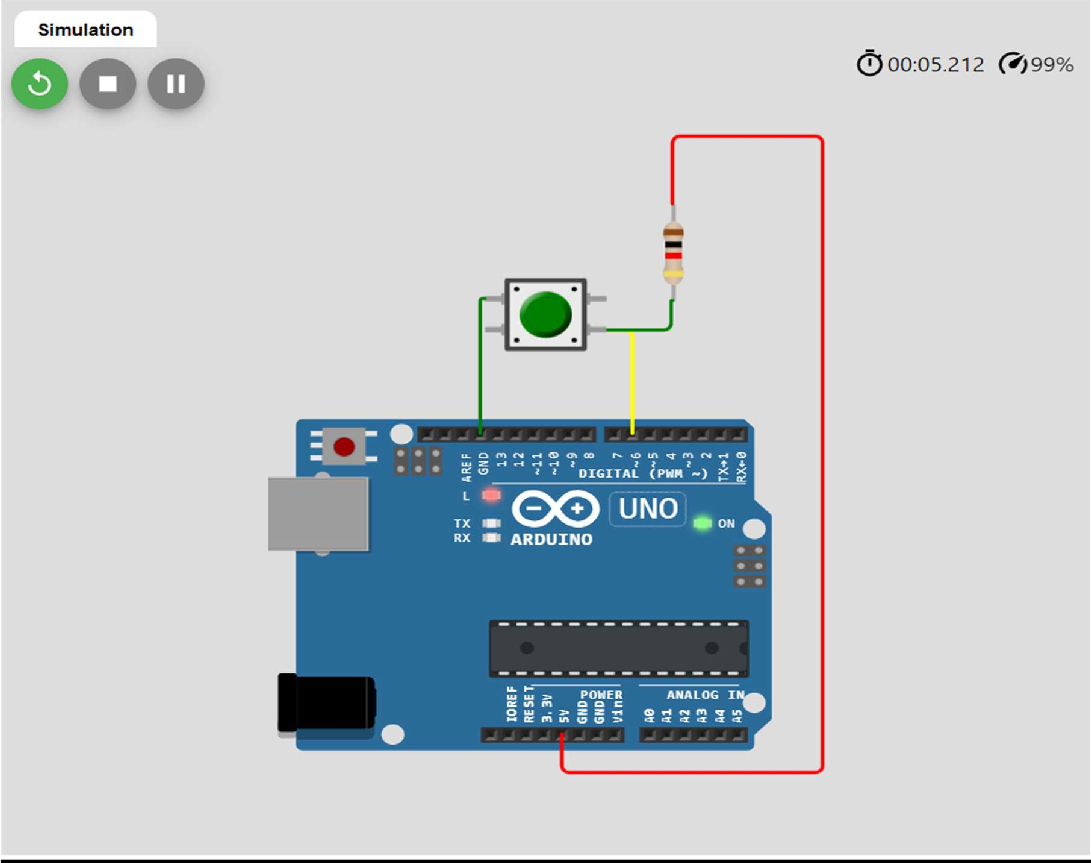
}

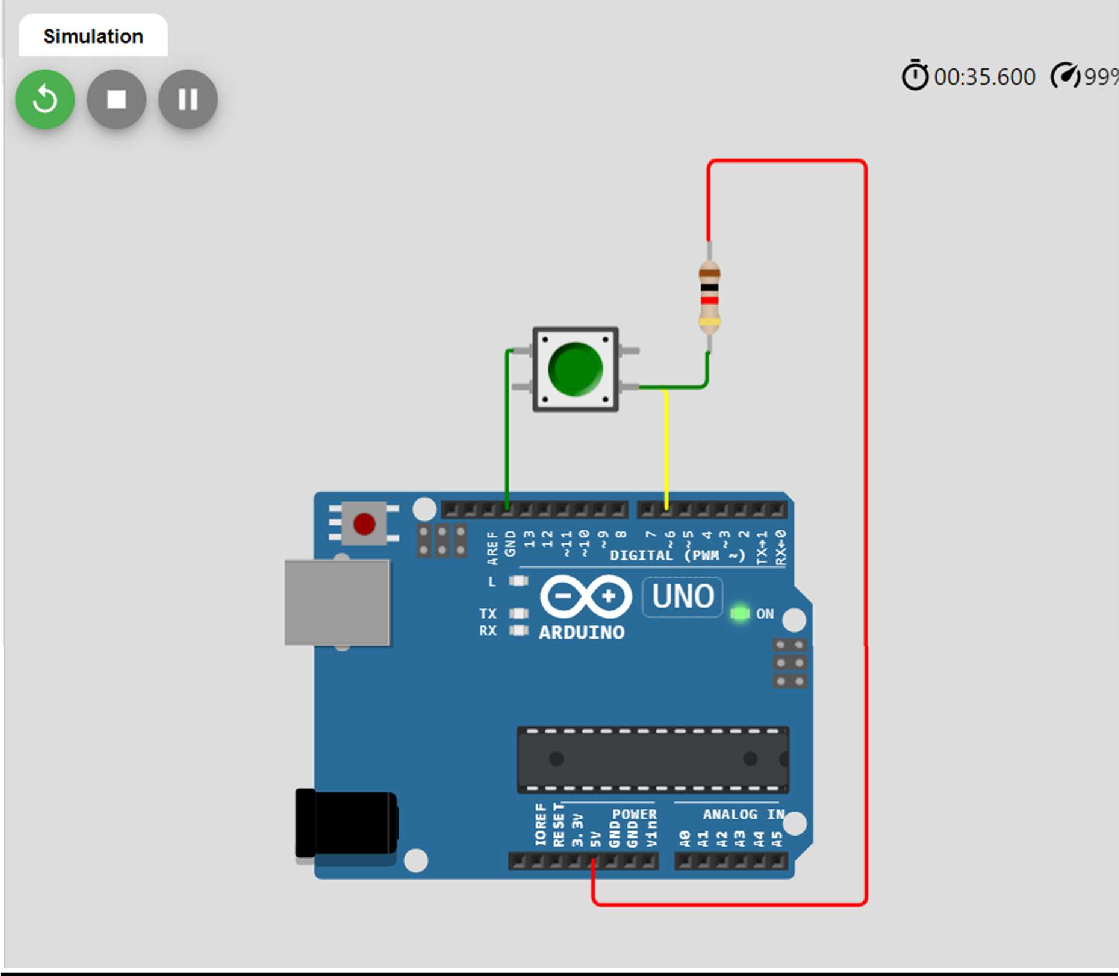
void loop() {

digitalWrite(13, digitalRead(6));

}

**OUTPUT:-**





## Task :-04

### Aim:-To interface DHT22 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings.

**Apparatus :-** Beadboard , Jumper Wires , DHT22 sensor, Arduino UNO board , Data cable and Arduino IDE software.

**Introduction:-** The DHT-22 is a digital-output relative humidity and temperature sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air.

#### Technical details:

* Power: 3-5V
* Current: 2.5mA
* Humidity: 20-95%, ±5% accuracy
* Temperature: 0 to 50°C, ±2°C accuracy

#### CODE :-

#include <DHT.h> #define DHTPIN 3

#define DHTTYPE DHT22

DHT dhtObject(DHTPIN,DHTTYPE);

void setup()

{

**Serial**.begin(9600); **Serial**.println("DHT initialized"); dhtObject.begin();

}

void loop() {

float humidity=dhtObject.readHumidity();

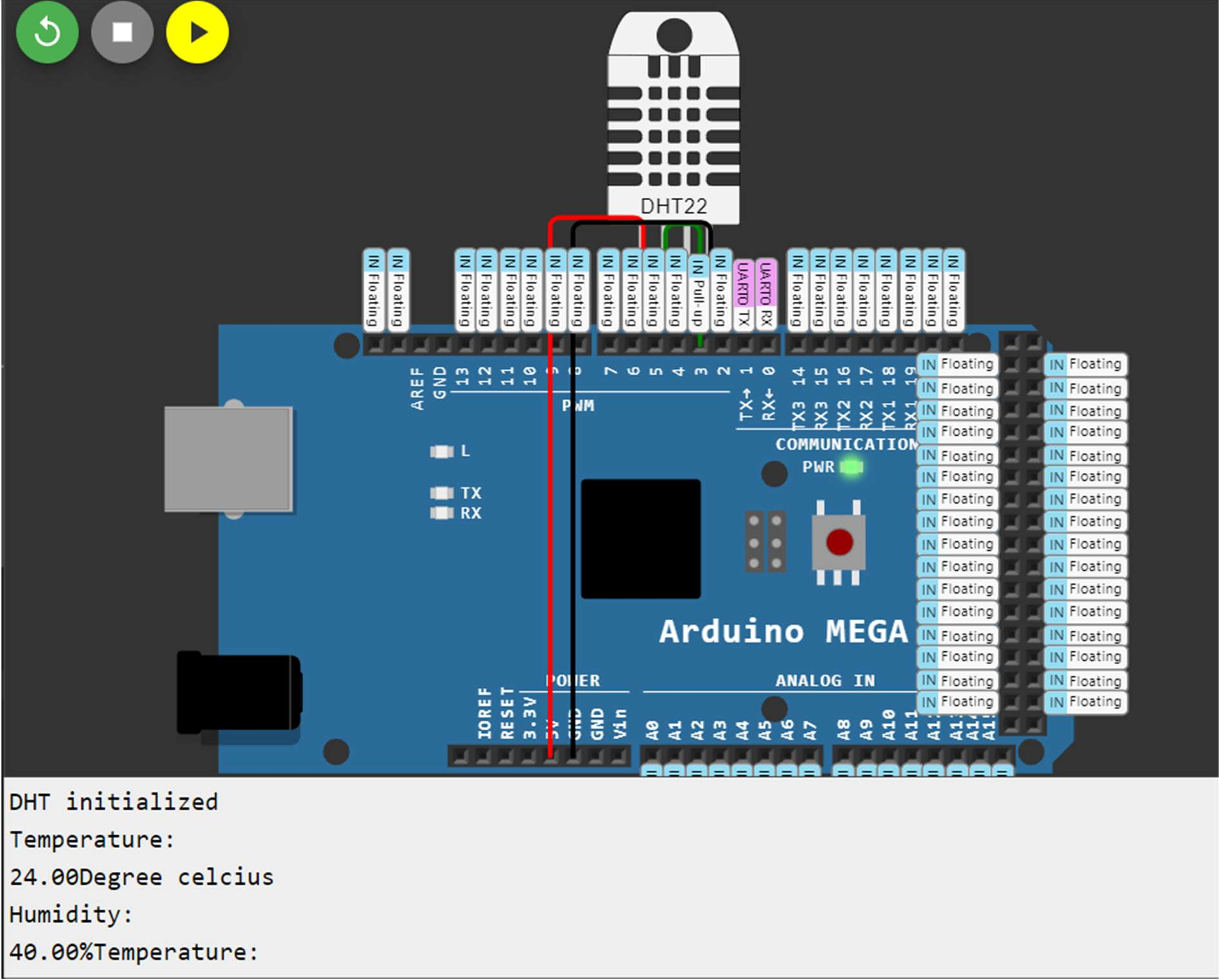
float Temperature=dhtObject.readTemperature();

**Serial**.println("Temperature:");

**Serial**.print(Temperature); **Serial**.print("Degree celcius"); **Serial**.println("\nHumidity:"); **Serial**.print(humidity); **Serial**.print("%"); delay(2000);

}

**OUTPUT:-**



## Task:-6

### Aim-To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.

* **Apparatus :** Arduino board,USB cable for uploading the code,DHT22 temperature,humidity sensor,SSD1306OLED display , jumper wires, breadboard
* **Introduction:** The Arduino will read temoerature and humidity values from DHT22 and show on OLED

CODE :

#include "DHT.h"

#include <Adafruit\_SSD1306.h> #include <Adafruit\_GFX.h> #define DHTPIN 2

#define OLED\_WIDTH 128

#define OLED\_HEIGHT 64 #define OLED\_ADDR 0x3C DHT dht(DHTPIN,DHTTYPE);

Adafruit\_SSD1306 display(OLED\_WIDTH,OLED\_HEIGHT);

void setup(){

float temperature =dht.readTemperature(); float Humidity=dht.readHumidity(); serial.begin(115200);

display.begin(SSD1306\_SWITCHCAPVCC,OLED\_ADDR); display.clearDisplay();

display.setTextSize(1.5); display.setTextColor(WHITE); display.println("Humidity:"); diplay.print(serial.println(humidity));

diplay.println(F("%")); display.setTextSize(1.5); display.setTextColor(WHITE); diplay.setCursor(0,17); display.println("Temperature:"); diplay.print(serial.println(temperature)); diplay.println(F("deg.Cel")); dht.begin();

diplay.diplay();

}

void loop(){ return 0;} **OUTPUT:-**

